**pwd**

pwd stands for “print working directory”. It outputs the name of the directory you are currently in, called the *working directory*.

**cd**

cd stands for “change directory”. Just as you would click on a folder in Windows Explorer or Finder, cd switches you into the directory you specify. In other words, cd changes the working directory.

**cd ..**

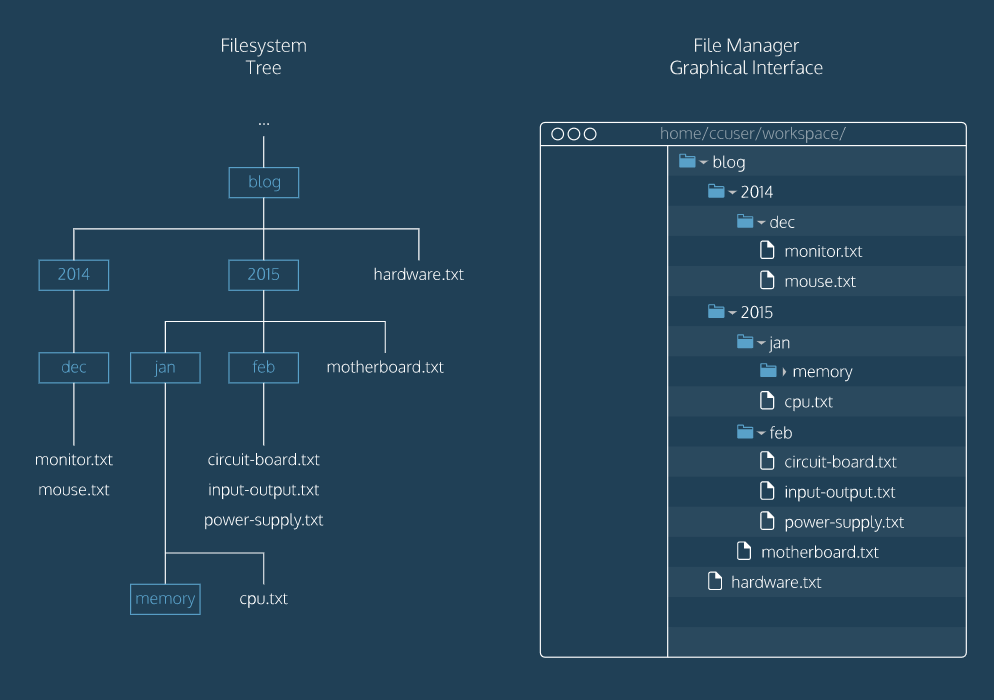
To move up one directory, use cd ..

**mkdir**

The mkdir command stands for “make directory”. It takes in a directory name as an argument, and then creates a new directory in the current working directory.

**touch**

The touch command creates a new file inside the working directory. It takes in a filename as an argument, and then creates an empty file in the current working directory.



* The command line is a text interface for the computer’s operating system. To access the command line, we use the terminal.
* A filesystem organizes a computer’s files and directories into a tree structure. It starts with the root directory. Each parent directory can contain more child directories and files.
* From the command line, you can navigate through files and folders on your computer:
  + pwd outputs the name of the current working directory.
  + ls lists all files and directories in the working directory.
  + cd switches you into the directory you specify.
  + mkdir creates a new directory in the working directory.
  + touch creates a new file inside the working directory.

**ls -i**

[-adfhktwW] [section] [-M path] [-P pager] [-S list]

[-m system] [-p string] name ...

a : find all matching entries

c : do not use cat file

d : print gobs of debugging information

D : as for -d, but also display the pages

f : same as whatis(1)

h : print this help message

k : same as apropos(1)

K : search for a string in all pages

t : use troff to format pages for printing

w : print location of man page(s) that would be displayed

(if no name given: print directories that would be searched)

W : as for -w, but display filenames only

C file : use `file' as configuration file

M path : set search path for manual pages to `path'

P pager : use program `pager' to display pages

S list : colon separated section list

m system : search for alternate system's man pages

p string : string tells which preprocessors to run

e - [n]eqn(1) p - pic(1) t - tbl(1)

g - grap(1) r - refer(1) v - vgrind(1)

**ls -l**

[-adfhktwW] [section] [-M path] [-P pager] [-S list]

[-m system] [-p string] name ...

a : find all matching entries

c : do not use cat file

d : print gobs of debugging information

D : as for -d, but also display the pages

f : same as whatis(1)

h : print this help message

k : same as apropos(1)

K : search for a string in all pages

t : use troff to format pages for printing

w : print location of man page(s) that would be displayed

(if no name given: print directories that would be searched)

W : as for -w, but display filenames only

C file : use `file' as configuration file

M path : set search path for manual pages to `path'

P pager : use program `pager' to display pages

S list : colon separated section list

m system : search for alternate system's man pages

p string : string tells which preprocessors to run

e - [n]eqn(1) p - pic(1) t - tbl(1)

g - grap(1) r - refer(1) v - vgrind(1)

The **echo** command accepts the string “Hello” as *standard input*, and echoes the string “Hello” back to the terminal as *standard output*.

Let’s learn more about standard input, standard output, and standard error:

1. *standard input*, abbreviated as **stdin**, is information inputted into the terminal through the keyboard or input device.
2. *standard output*, abbreviated as **stdout**, is the information outputted after a process is run.
3. *standard error*, abbreviated as **stderr**, is an error message outputted by a failed process.

Redirection reroutes standard input, standard output, and standard error to or from a different location.

**cat** -- concatenate and print files

**cat** [**-benstuv**] [file ...]

The **cat** utility reads files sequentially, writing them to the standard

output. The file operands are processed in command-line order. If file

is a single dash (`-') or absent, **cat** reads from the standard input. If

file is a UNIX domain socket, **cat** connects to it and then reads it until

EOF. This complements the UNIX domain binding capability available in

inetd(8).

The options are as follows:

**-b** Number the non-blank output lines, starting at 1.

**-e** Display non-printing characters (see the **-v** option), and display

a dollar sign (`$') at the end of each line.

# |

| is a “pipe”. The | takes the standard output of the command on the left, and pipes it as standard input to the command on the right. You can think of this as “command to command” redirection.

Multiple | ‘s can be chained together. Here the standard output of cat volcanoes.txt is “piped” to the wc command. The standard output of wc is then “piped” to cat. Finally, the standard output of cat is redirected to islands.txt.

# sort

sort takes the standard input and orders it alphabetically for the standard output.

**uniq**

uniq stands for “unique” and filters out adjacent, duplicate lines in a file. A more effective way to call uniqis to call sort to alphabetize a file, and “pipe” the standard output to uniq.

GREP(1) BSD General Commands Manual GREP(1)

NNAAMMEE

ggrreepp, eeggrreepp, ffggrreepp, zzggrreepp, zzeeggrreepp, zzffggrreepp -- file pattern searcher

SSYYNNOOPPSSIISS

ggrreepp [--aabbccddDDEEFFGGHHhhIIiiJJLLllmmnnOOooppqqRRSSssUUVVvvwwxxZZ] [--AA \_n\_u\_m] [--BB \_n\_u\_m] [--CC[\_n\_u\_m]]

[--ee \_p\_a\_t\_t\_e\_r\_n] [--ff \_f\_i\_l\_e] [----bbiinnaarryy--ffiilleess=\_v\_a\_l\_u\_e] [----ccoolloorr[=\_w\_h\_e\_n]]

[----ccoolloouurr[=\_w\_h\_e\_n]] [----ccoonntteexxtt[=\_n\_u\_m]] [----llaabbeell] [----lliinnee--bbuuffffeerreedd]

[----nnuullll] [\_p\_a\_t\_t\_e\_r\_n] [\_f\_i\_l\_e \_.\_.\_.]

DDEESSCCRRIIPPTTIIOONN

The ggrreepp utility searches any given input files, selecting lines that

match one or more patterns. By default, a pattern matches an input line

if the regular expression (RE) in the pattern matches the input line

without its trailing newline. An empty expression matches every line.

Each input line that matches at least one of the patterns is written to

the standard output.

ggrreepp is used for simple patterns and basic regular expressions (BREs);

eeggrreepp can handle extended regular expressions (EREs). See re\_format(7)

for more information on regular expressions. ffggrreepp is quicker than both

ggrreepp and eeggrreepp, but can only handle fixed patterns (i.e. it does not

interpret regular expressions). Patterns may consist of one or more

lines, allowing any of the pattern lines to match a portion of the input.

zzggrreepp, zzeeggrreepp, and zzffggrreepp act like ggrreepp, eeggrreepp, and ffggrreepp, respectively,

but accept input files compressed with the compress(1) or gzip(1) com-

pression utilities.

The following options are available:

--AA \_n\_u\_m, ----aafftteerr--ccoonntteexxtt=\_n\_u\_m

Print \_n\_u\_m lines of trailing context after each match. See also

the --BB and --CC options.

--aa, ----tteexxtt

Treat all files as ASCII text. Normally ggrreepp will simply print

``Binary file ... matches'' if files contain binary characters.

Use of this option forces ggrreepp to output lines matching the spec-

ified pattern.

--BB \_n\_u\_m, ----bbeeffoorree--ccoonntteexxtt=\_n\_u\_m

Print \_n\_u\_m lines of leading context before each match. See also

the --AA and --CC options.

--bb, ----bbyyttee--ooffffsseett

The offset in bytes of a matched pattern is displayed in front of

the respective matched line.

--CC[\_n\_u\_m, ----ccoonntteexxtt=\_n\_u\_m]

Print \_n\_u\_m lines of leading and trailing context surrounding each

match. The default is 2 and is equivalent to --AA \_2 --BB \_2. Note:

no whitespace may be given between the option and its argument.

--cc, ----ccoouunntt

Only a count of selected lines is written to standard output.

----ccoolloouurr=[\_w\_h\_e\_n, ----ccoolloorr=[\_w\_h\_e\_n]]

Mark up the matching text with the expression stored in

GREP\_COLOR environment variable. The possible values of when can

be `never', `always' or `auto'.

--DD \_a\_c\_t\_i\_o\_n, ----ddeevviicceess=\_a\_c\_t\_i\_o\_n

Specify the demanded action for devices, FIFOs and sockets. The

default action is `read', which means, that they are read as if

they were normal files. If the action is set to `skip', devices

will be silently skipped.

--dd \_a\_c\_t\_i\_o\_n, ----ddiirreeccttoorriieess=\_a\_c\_t\_i\_o\_n

Specify the demanded action for directories. It is `read' by

default, which means that the directories are read in the same

manner as normal files. Other possible values are `skip' to

silently ignore the directories, and `recurse' to read them

recursively, which has the same effect as the --RR and --rr option.

--EE, ----eexxtteennddeedd--rreeggeexxpp

Interpret \_p\_a\_t\_t\_e\_r\_n as an extended regular expression (i.e. force

ggrreepp to behave as eeggrreepp).

--ee \_p\_a\_t\_t\_e\_r\_n, ----rreeggeexxpp=\_p\_a\_t\_t\_e\_r\_n

Specify a pattern used during the search of the input: an input

line is selected if it matches any of the specified patterns.

This option is most useful when multiple --ee options are used to

specify multiple patterns, or when a pattern begins with a dash

(`-').

----eexxcclluuddee

If specified, it excludes files matching the given filename pat-

tern from the search. Note that ----eexxcclluuddee patterns take priority

over ----iinncclluuddee patterns, and if no ----iinncclluuddee pattern is speci-

fied, all files are searched that are not excluded. Patterns are

matched to the full path specified, not only to the filename com-

ponent.

----eexxcclluuddee--ddiirr

If --RR is specified, it excludes directories matching the given

filename pattern from the search. Note that ----eexxcclluuddee--ddiirr pat-

terns take priority over ----iinncclluuddee--ddiirr patterns, and if no

----iinncclluuddee--ddiirr pattern is specified, all directories are searched

that are not excluded.

--FF, ----ffiixxeedd--ssttrriinnggss

Interpret \_p\_a\_t\_t\_e\_r\_n as a set of fixed strings (i.e. force ggrreepp to

behave as ffggrreepp).

--ff \_f\_i\_l\_e, ----ffiillee=\_f\_i\_l\_e

Read one or more newline separated patterns from \_f\_i\_l\_e. Empty

pattern lines match every input line. Newlines are not consid-

ered part of a pattern. If \_f\_i\_l\_e is empty, nothing is matched.

--GG, ----bbaassiicc--rreeggeexxpp

Interpret \_p\_a\_t\_t\_e\_r\_n as a basic regular expression (i.e. force ggrreepp

to behave as traditional ggrreepp).

--HH Always print filename headers with output lines.

--hh, ----nnoo--ffiilleennaammee

Never print filename headers (i.e. filenames) with output lines.

----hheellpp Print a brief help message.

--II Ignore binary files. This option is equivalent to

----bbiinnaarryy--ffiillee=\_w\_i\_t\_h\_o\_u\_t\_-\_m\_a\_t\_c\_h option.

--ii, ----iiggnnoorree--ccaassee

Perform case insensitive matching. By default, ggrreepp is case sen-

sitive.

----iinncclluuddee

If specified, only files matching the given filename pattern are

searched. Note that ----eexxcclluuddee patterns take priority over

----iinncclluuddee patterns. Patterns are matched to the full path speci-

fied, not only to the filename component.

----iinncclluuddee--ddiirr

If --RR is specified, only directories matching the given filename

pattern are searched. Note that ----eexxcclluuddee--ddiirr patterns take pri-

ority over ----iinncclluuddee--ddiirr patterns.

--JJ,, ----bbzz22ddeeccoommpprreessss

Decompress the bzip2(1) compressed file before looking for the

text.

--LL, ----ffiilleess--wwiitthhoouutt--mmaattcchh

Only the names of files not containing selected lines are written

to standard output. Pathnames are listed once per file searched.

If the standard input is searched, the string ``(standard

input)'' is written.

--ll, ----ffiilleess--wwiitthh--mmaattcchheess

Only the names of files containing selected lines are written to

standard output. ggrreepp will only search a file until a match has

been found, making searches potentially less expensive. Path-

names are listed once per file searched. If the standard input

is searched, the string ``(standard input)'' is written.

----mmmmaapp Use mmap(2) instead of read(2) to read input, which can result in

better performance under some circumstances but can cause unde-

fined behaviour.

--mm \_n\_u\_m\_, ----mmaaxx--ccoouunntt=\_n\_u\_m

Stop reading the file after \_n\_u\_m matches.

--nn, ----lliinnee--nnuummbbeerr

Each output line is preceded by its relative line number in the

file, starting at line 1. The line number counter is reset for

each file processed. This option is ignored if --cc, --LL, --ll, or --qq

is specified.

----nnuullll Prints a zero-byte after the file name.

--OO If --RR is specified, follow symbolic links only if they were

explicitly listed on the command line. The default is not to

follow symbolic links.

--oo,, ----oonnllyy--mmaattcchhiinngg

Prints only the matching part of the lines.

--pp If --RR is specified, no symbolic links are followed. This is the

default.

--qq, ----qquuiieett, ----ssiilleenntt

Quiet mode: suppress normal output. ggrreepp will only search a file

until a match has been found, making searches potentially less

expensive.

--RR, --rr, ----rreeccuurrssiivvee

Recursively search subdirectories listed.

--SS If --RR is specified, all symbolic links are followed. The default

is not to follow symbolic links.

--ss, ----nnoo--mmeessssaaggeess

Silent mode. Nonexistent and unreadable files are ignored (i.e.

their error messages are suppressed).

--UU, ----bbiinnaarryy

Search binary files, but do not attempt to print them.

--VV, ----vveerrssiioonn

Display version information and exit.

--vv, ----iinnvveerrtt--mmaattcchh

Selected lines are those \_n\_o\_t matching any of the specified pat-

terns.

--ww, ----wwoorrdd--rreeggeexxpp

The expression is searched for as a word (as if surrounded by

`[[:<:]]' and `[[:>:]]'; see re\_format(7)).

--xx, ----lliinnee--rreeggeexxpp

Only input lines selected against an entire fixed string or regu-

lar expression are considered to be matching lines.

--yy Equivalent to --ii. Obsoleted.

--ZZ, --zz, ----ddeeccoommpprreessss

Force ggrreepp to behave as zzggrreepp.

----bbiinnaarryy--ffiilleess=\_v\_a\_l\_u\_e

Controls searching and printing of binary files. Options are

\_b\_i\_n\_a\_r\_y, the default: search binary files but do not print them;

\_w\_i\_t\_h\_o\_u\_t\_-\_m\_a\_t\_c\_h: do not search binary files; and \_t\_e\_x\_t: treat all

files as text.

----ccoonntteexxtt[=\_n\_u\_m]

Print \_n\_u\_m lines of leading and trailing context. The default is

2.

----lliinnee--bbuuffffeerreedd

Force output to be line buffered. By default, output is line

buffered when standard output is a terminal and block buffered

otherwise.

If no file arguments are specified, the standard input is used.

EENNVVIIRROONNMMEENNTT

GREP\_OPTIONS May be used to specify default options that will be placed

at the beginning of the argument list. Backslash-escaping

is not supported, unlike the behavior in GNU grep.

EEXXIITT SSTTAATTUUSS

The ggrreepp utility exits with one of the following values:

0 One or more lines were selected.

1 No lines were selected.

>1 An error occurred.

EEXXAAMMPPLLEESS

To find all occurrences of the word `patricia' in a file:

$ grep 'patricia' myfile

To find all occurrences of the pattern `.Pp' at the beginning of a line:

$ grep '^\.Pp' myfile

The apostrophes ensure the entire expression is evaluated by ggrreepp instead

of by the user's shell. The caret `^' matches the null string at the

beginning of a line, and the `\' escapes the `.', which would otherwise

match any character.

To find all lines in a file which do not contain the words `foo' or

`bar':

$ grep -v -e 'foo' -e 'bar' myfile

A simple example of an extended regular expression:

$ egrep '19|20|25' calendar

Peruses the file `calendar' looking for either 19, 20, or 25.

SSEEEE AALLSSOO

ed(1), ex(1), gzip(1), sed(1), re\_format(7)

SSTTAANNDDAARRDDSS

The ggrreepp utility is compliant with the IEEE Std 1003.1-2008 (``POSIX.1'')

specification.

The flags [--AAaaBBbbCCDDddGGHHhhIIJJLLmmooPPRRSSUUVVwwZZ] are extensions to that specification,

and the behaviour of the --ff flag when used with an empty pattern file is

left undefined.

All long options are provided for compatibility with GNU versions of this

utility.

Historic versions of the ggrreepp utility also supported the flags [--rruuyy].

This implementation supports those options; however, their use is

strongly discouraged.

HHIISSTTOORRYY

The ggrreepp command first appeared in Version 6 AT&T UNIX.

BBUUGGSS

The ggrreepp utility does not normalize Unicode input, so a pattern contain-

ing composed characters will not match decomposed input, and vice versa.

BSD July 28, 2010 BSD

**grep I**

grep stands for “*global regular expression print*”. It searches files for lines that match a pattern and returns the results. It is also case sensitive.

**grep -i**

enables the command to be case insensitive.

grep II

grep -R searches all files in a directory and outputs filenames and lines containing matched results. -R stands for “recursive”. grep -Rl searches all files in a directory and outputs only filenames with matched results. -R stands for “recursive” and l stands for “files with matches”.

**sed**

**sed stands for “stream editor”. It accepts standard input and modifies it based on an expression, before displaying it as output data. It is similar to “find and replace”. s: stands for “substitution”. it is always used when using sed for substitution.**

> redirects standard output of a command to a file, overwriting previous content.

>> redirects standard output of a command to a file, appending new content to old content.

< redirects standard input to a command.

| redirects standard output of a command to another command.

A number of other commands are powerful when combined with redirection commands:

sort: sorts lines of text alphabetically.

uniq: filters duplicate, adjacent lines of text.

grep: searches for a text pattern and outputs it.

sed : searches for a text pattern, modifies it, and outputs it.

Show hidden ("dot") files with ls -a

Show file details with ls -l

Combine multiple flags like ls -l -a

You can sometimes chain flags like ls -la instead of ls -l -a

head outputs the first few lines of a file. The -n flag specifies the number of lines to show (the default is 10):